TP1 Analyse, Traitement de l'Image et Vision

Master ID3D - october 2020

Yann-Situ GAZULL
Master IF
ENS de Lyon

Matheo DUMONT
Master ID3D
Université Claude Bernard Lyon 1

Abstract

This document is a homework done by Yann-Situ Gazull and Matheo Dumont for the course "Analyse, Traitement de l'Image et Vision" given by Saida Bouakaz.

The source code can be found on Github here https://github.com/MatheoDumont/ATIV_tp1.

Index Terms

contour, convolution, image, processing, analyse, rhododendron

CONTENTS

I. Contour detection with convolution

A. Computing gradient

To compute the gradient, we use the convolution product, to apply a kernel h to a base image I. Depending on the kernel, we can obtain different results.

The convolution product can be define like this, for the point (x, y).

$$(I * h)(x, y) = \int \int I(x, y)h(x - u, y, -v)dudv$$

B. Computing bidirectional gradient amplitude and angle

The bidirectional gradient is computed using the horizontal called here H and vertical axis called here V, applying two kernel, one for each direction, to the base image.

$$H = 1/3 \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}$$

$$V = 1/3 \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}$$

By extension, we will call the result of $I * H$ and $I * V$ respectively H and V. Once we have applied those two kernel to I, we can compute the amplitude of gradient with different norms. The two most interesting norms to compute the amplitude are respectively the l_∞ and the l_2 (euclidean) norm which gives :

$$A_\infty(x, y) = \max\left(H(x, y), V(x, y)\right)$$
$$A_2(x, y) = \sqrt{H(x, y)^2 + V(x, y)^2}$$

Notice that we have to divide A_2 by $\sqrt{2}$ in order to have $0 \leq A_2 \leq 255$. We chose A_∞ since it yields the best results.

The angle $\theta_{bidir}(x, y)$ can be compute with :

$$\theta_{bidir}(x, y) = sign\left(H(x, y)\right) \arctan\left(\frac{V(x, y)}{H(x, y)}\right)$$
$$= atan2(V(x, y), H(x, y))$$

Where $sign(x)$ equals to 1 if $x > 0$, -1 if $x < 0$ and 0 otherwise.

C. Computing multi-directional gradient amplitude and angle

The multi-directional gradient is computed using four kernels, H and V previously defined plus two other, the first one for the direction $\frac{\pi}{4}$ called R and the second one for the direction $\frac{3\pi}{4}$ called L:

$$R = 1/3 \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}$$

$$L = 1/3 \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix}$$

By extension, we will call the result of $I * R$ and $I * L$ respectively R and L. For computing the gradient amplitude in this case, we simply take the maximum value between the four direction : $\{H, V, L, R\}$ for each pixel : $A(x, y) = \max(H(x, y), V(x, y), L(x, y), R(x, y)$.

For computing the angle, we can first notice that each kernel can be used not for one but two directions, considering the case when the amplitude is above or below zero. Then, we have with $\{H, V, L, R\}$, not four but eight directions, which is an 8-adjacency in an image.

Then, the angle $\theta(x, y)$ for one pixel (x, y) can be computed as :

$$G = [H, R, V, L] = [G_0, G_1, G_2, G_3]$$
$$i(x, y) = argmax_{k \in \{0,1,2,3\}}(G_k(x, y))$$
$$\theta(x, y) = i(x, y) \text{ if } G_{i(x,y)}(x, y) > 0 \text{ else } i(x, y) + 4$$

Hence, the angle is an integer that is between 0 and 7.

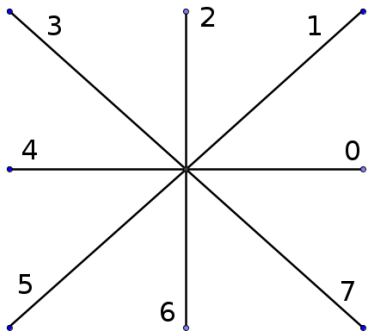


Fig. 1: Integer angles returned by our method θ.

Fig. 2: Bidirectional gradient amplitude on *Lena* computed using A_∞ (first image) and A_2 (second image).

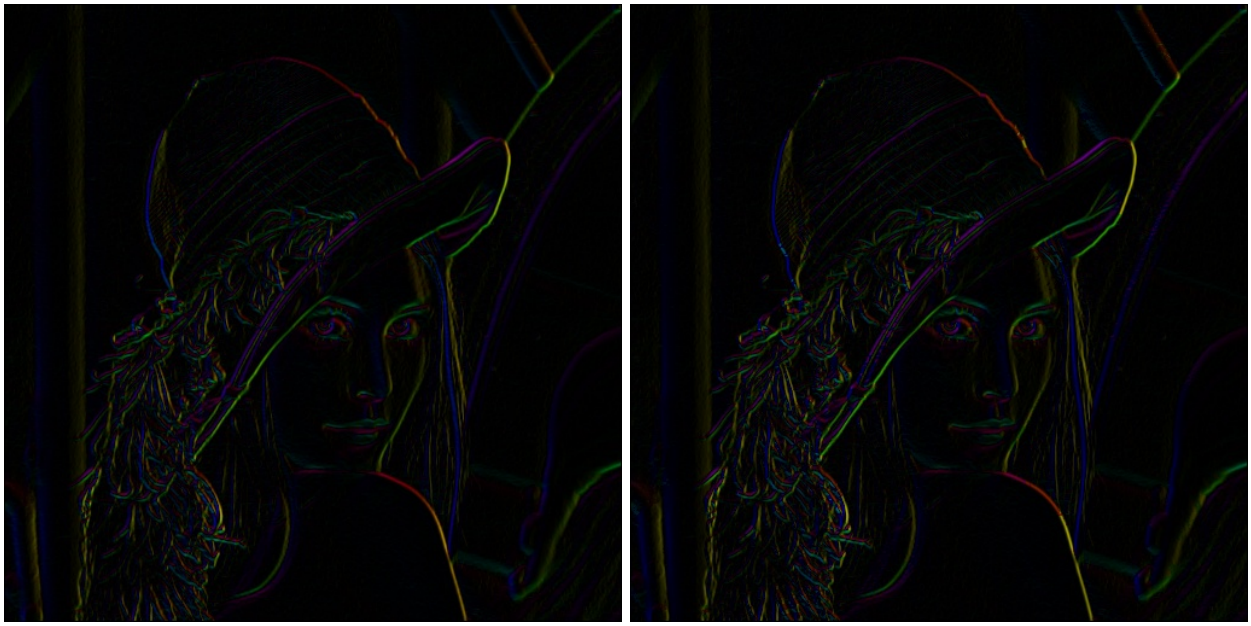


Fig. 3: Bidirectional and multi-directional gradients angles $\theta_{bidir}(x, y)$ and $\theta(x, y)$ on *Lena*, displayed using HSV color format : $H := \theta$, $S := 255$ and $V := A_\infty$.

II. Contour thresholding

We implemented several methods in order to decide if a pixel is on a contour or not from the amplitude and angle maps. The first two are the global thresholding method and the hysterisis thresholding method. Another approach is explained in the section IV-A.

A. Global thresholding

In this method, we decide that a pixel is on a contour if and only if the amplitude of its gradient is above a certain threshold t_{max}. The main advantages is that it is easy to implement and to understand. The main drawbacks are that a soft contour (for example due to anti-aliasing or soft blurring) might be discarded and that it is not strong against noise.

Fig. 4: Thresholding of the image given by a global threshold with 0.1.

B. Hysteresis thresholding

In this method, we improve the global thresholding method by introducing another threshold value $t_{min} < t_{max}$ The idea is to filter the image two times. We find ourselves with 3 cases:

1) For all the pixels below t_{min}, we consider them as not part of the contour.
2) For all the pixels above t_{max}, we consider them as part of the contour.
3) For all the pixels p such that $t_{min} < A(p) < t_{max}$, we consider them as part of the contour only if they have in their neighbouring a pixel above t_{max}.

This way, we obtain a binary image, with the pixels for the contour at 1 and the others at 0. A drawback of this method is that it can be hard to choose appropriate t_{min} and t_{max} given an image.

Fig. 5: Thresholding of the image given by the hysteresis method with $t_{min} = 0.07$, $t_{max} = 0.16$ and a 8-adjacency neighbouring.

III. Post-processing

A. Contour refinement

The contour computed on an image add some padding to the original image around the contour, the idea is to take advantage of it to refine the contour, by choosing the one pixel around a contour that maximize its gradient with value one and the others with value zero. The refinement algorithm is presented in algorithm 1 : $affinage\_max\_loc(I,\ A,\ \theta)$.

Algorithm 1 $affinage\_max\_loc(I,\ A,\ \theta)$

Require: amplitude A and gradient angle θ already computed
Ensure: refine binary contour image I
  $contour$ initialized with zeros
  for $(x, y) \in I$ **do**
    if $I(x, y) < 0.5$ **then**
      continue, this isn't a contour
    else
      $\overrightarrow{steep} \leftarrow pos\_from\_angle(\theta(x, y))$
      $\overrightarrow{inv} \leftarrow pos\_from\_angle(-\theta(x, y))$
      $points = [(x, y) + inv, (x, y), (x, y) + steep]$
      $G = [A((x, y) + inv),\ A(x, y),\ A((x, y) + steep)]$
      $i = argmax_{k\in\{0,1,2\}}(G_k(x, y))$
      $contour(points[i]) \leftarrow 1$
    end if
  end for
  return $contour$ of size I, filled with zeros and ones

In the algorithm 1, $pos\_from\_angle(\theta)$ returns the vector of norm$_\infty$ equal to 1 that corresponds to the integer angle θ : for instance if $\theta = 7$, it returns $(1, -1)$ (see figure 1).

This method works well if we have contours that have a thickness lower than 3 pixels, but otherwise, this can lead to

the creation of two or more parallel contours of thickness equal to 1 pixel.

Fig. 6: Refinement of the image in figure 6.

B. Contour closing

Firstly, we tried to apply a dilatation then an erosion to our contour with an isotropic structuring element.

Fig. 7: Dilatation then erosion (with a 3×3 structuring element) of the contour obtained in figure 4.

The results were not convincing (see figure 7), so we tried a smarter approach that we called the *gradient dilatation method* :

it consists in dilating each pixel in the direction perpendicular to its gradient within a chosen radius, in order to extrapolate a contour. However we have encountered some issues because of the integer directions of the gradient. The figure 8 illustrates well this issue : the triangle bottom contour has an integer angle slope close to 1 but it is dilated quite randomly in the directions 0, 1 and 2. Some other results of the gradient dilatation are shown in figure 9.

IV. Other contour detection methods

A. Contour path research

The contour path research method is quite different from the other methods. As the hysteresis method, this method uses two different thresholds t_{min} and t_{max} with $t_{min} < t_{max}$. However, in this method we try to follow connected contours. To realize this task we call a recursive function *path_contour_pix* from each pixel (x, y) that satisfies $A(x, y) > t_{max}$.

The recursive function *path_contour_pix*(x, y) updates the contour binary map (which is initialized with zeros) by searching in a direction perpendicular to the gradient and creating a path included in the contour :

Algorithm 2 $path\_contour\_pix(x,\ y,\ \theta,\ A,\ contour)$

Require: amplitude A and gradient angle θ already computed
Ensure: update the binary map $contour$ recursively
  $V \leftarrow PotentialNext(x, y, \theta(x, y))$
  $(xmax, ymax) \leftarrow (x, y)$
  $maxi \leftarrow 0.0$
  for $(x', y') \in V$ **do**
    if $A(x', y') > maxi$ **then**
      $(xmax, ymax) \leftarrow (x', y')$
      $maxi \leftarrow A(x', y')$
    end if
  end for
  if $maxi > t_{min}$ **then**
    $contour(xmax, ymax) \leftarrow 1$
    $path\_contour\_pix(xmax,\ ymax,\ \theta,\ A,\ contour)$
  else
    Do nothing : this is the end of a contour path
  end if

Where the function $PotentialNext(x, y, \theta(x, y))$ returns a 3-set of neighbouring pixels in the direction 90° after $\theta(x, y)$ in the counter-clockwise direction. For instance, if $\theta(x, y) = 1$, it returns $\{(x-1, y),\ (x-1, y+1),\ (x, y+1)\}$; if $\theta(x, y) = 6$, it returns $\{(x+1, y-1),\ (x+1, y),\ (x+1, y+1)\}$.

For some results on the path contour method, see figure 10.

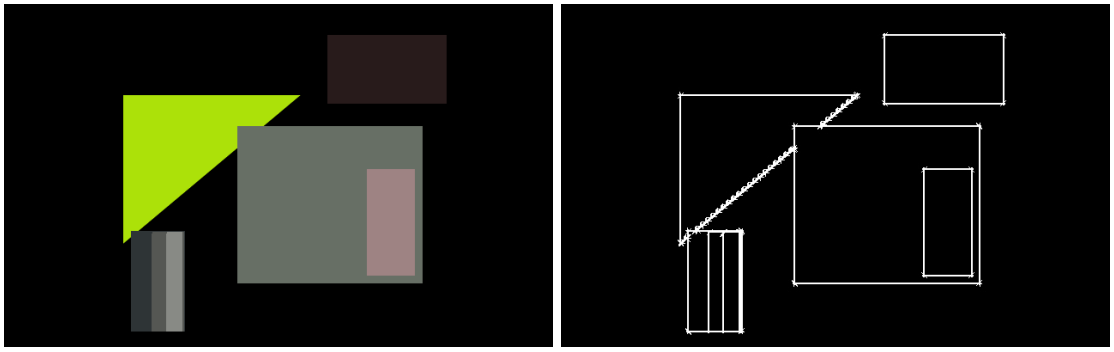


Fig. 8: Gradient dilatation method on a polygon figure.

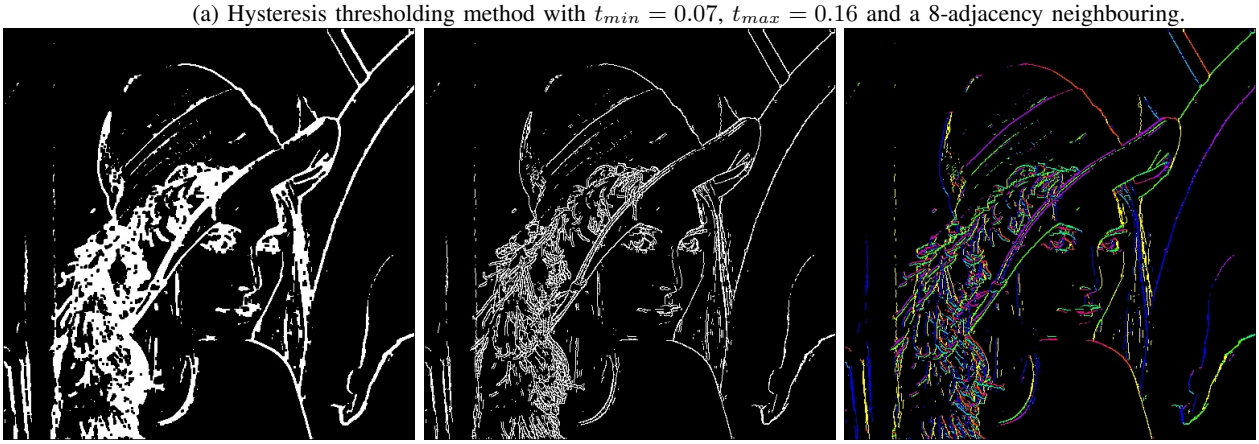
Fig. 9: Gradient dilatation method on *Lena* (from the contour in figure 4) with radius 1 and radius 3.

Fig. 10: Path contour method on *Lena*. The first contour image was obtained with $t_{min} = 0.025$ and $t_{max} = 0.12$, the second with $t_{min} = 0.035$ and $t_{max} = 0.10$.

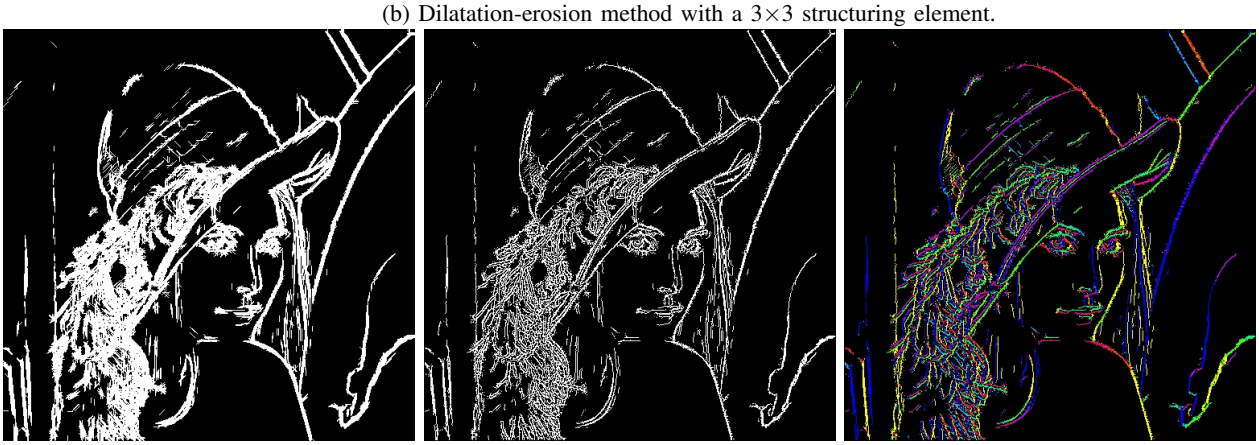
V. Comparison of results

It is interesting to compare the results of the different methods. More precisely, it is relevant to study the connectedness and the smoothness of the resulting contours.

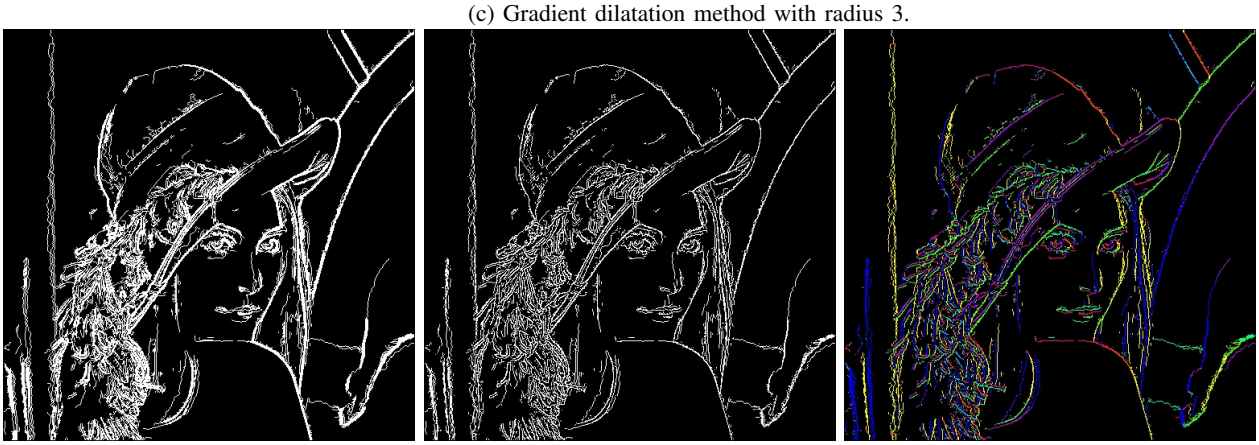
(a) Hysteresis thresholding method with $t_{min} = 0.07$, $t_{max} = 0.16$ and a 8-adjacency neighbouring.



(b) Dilatation-erosion method with a 3×3 structuring element.



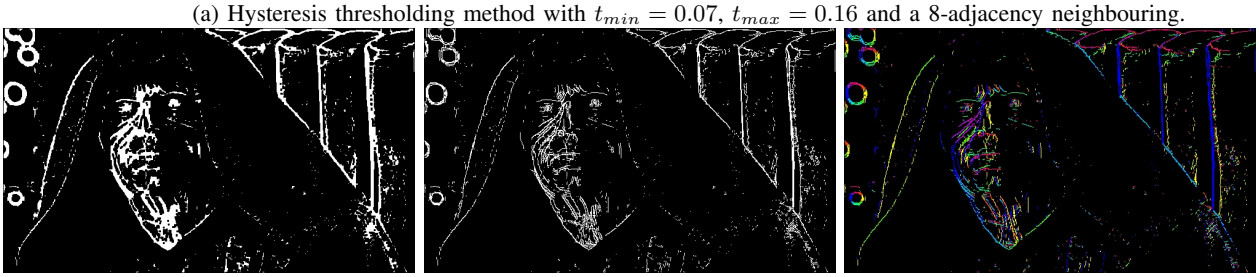
(c) Gradient dilatation method with radius 3.



(d) Path contour method with $t_{min} = 0.025$ and $t_{max} = 0.12$.

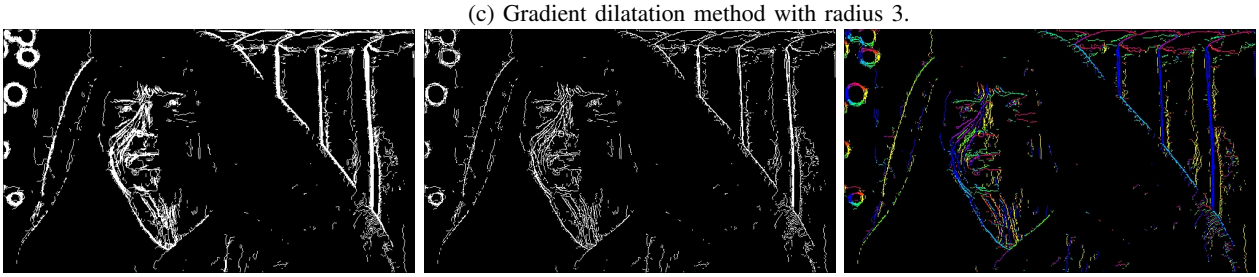
Fig. 11: Comparison of different methods on *Lena*, before refinement, after refinement and with color angles.

(a) Hysteresis thresholding method with $t_{min} = 0.07$, $t_{max} = 0.16$ and a 8-adjacency neighbouring.



(b) Dilatation-erosion method with a 3×3 structuring element.

(c) Gradient dilatation method with radius 3.



(d) Path contour method with $t_{min} = 0.025$ and $t_{max} = 0.12$.

Fig. 12: Comparison of different methods on *Palpa*, before refinement, after refinement and with color angles.